

UNCLASSIFIED

AD 408 414

DEFENSE DOCUMENTATION CENTER

FOR

SCIENTIFIC AND TECHNICAL INFORMATION

CAMERON STATION, ALEXANDRIA, VIRGINIA

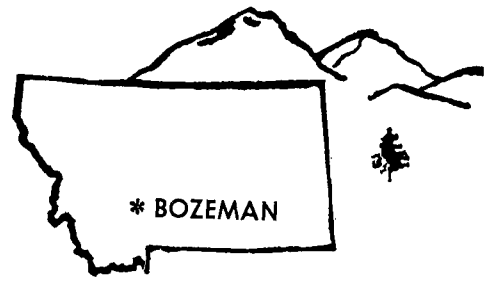


UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

408414

63-4-2



Scientific Report No. 4

AFCRL - 63 - 68

AF19(604) - 6619

**On Dynamic Switching in One-Dimensional
Iterative Logic Networks**

BY W. L. KILMER

March, 1963

Project 5632

Task 563202

ELECTRONICS RESEARCH LABORATORY

Endowment and Research Foundation

Montana State College

Bozeman, Montana

Prepared for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES

Office of Aerospace Research

United States Air Force

Bedford, Massachusetts

RECEIVED
JUL 13 1963
TISIA A

408 414

**electronics
research
laboratory**

ENDOWMENT AND RESEARCH FOUNDATION AT
MONTANA STATE COLLEGE, BOZEMAN, MONT.

"Requests for additional copies by Agencies of the Department of Defense, their contractors, and other Government agencies should be directed to the:

DEFENSE DOCUMENTATION CENTER (DDC)
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA

Department of Defense contractors must be established for ASTIA services or have their 'need-to-know' certified by the cognizant military agency of their project or contract."

"All other persons and organizations should apply to the:

U. S. DEPARTMENT OF COMMERCE
OFFICE OF TECHNICAL SERVICES
WASHINGTON 25, D. C. "

<p>AF Cambridge Research Laboratories, Bedford, Mass. ON DYNAMIC SWITCHING IN ONE-DIMENSIONAL ITERATIVE LOGIC NETWORKS, by W.L. Kilmer, March 1963. 26 pp. AFCRL-63-68</p> <p>The main purpose of this paper is to present some results on the problem of determining, for an arbitrary SITN cell definition, the various ways in which corresponding SITNs can switch from equilibrium to equilibrium, and equilibrium to cycle, following single x_i input value changes. These results are also applied to non-SITN models discussed earlier by the author in order to extend the present theory of switching dynamics for iterative systems.</p>	<p>UNCLASSIFIED</p> <ol style="list-style-type: none"> 1. Nonlinear Control Theory 2. Switching Theory 3. Mathematical Logic <p>I. Kilmer, W.L.</p>	<p>AF Cambridge Research Laboratories, Bedford, Mass. ON DYNAMIC SWITCHING IN ONE-DIMENSIONAL ITERATIVE LOGIC NETWORKS, by W.L. Kilmer, March 1963. 26 pp. AFCRL-63-68</p> <p>The main purpose of this paper is to present some results on the problem of determining, for an arbitrary SITN cell definition, the various ways in which corresponding SITNs can switch from equilibrium to equilibrium, and equilibrium to cycle, following single x_i input value changes. These results are also applied to non-SITN models discussed earlier by the author in order to extend the present theory of switching dynamics for iterative systems.</p>	<p>UNCLASSIFIED</p> <ol style="list-style-type: none"> 1. Nonlinear Control Theory 2. Switching Theory 3. Mathematical Logic <p>I. Kilmer, W.L.</p>	<p>UNCLASSIFIED</p>
<p>AF Cambridge Research Laboratories, Bedford, Mass. ON DYNAMIC SWITCHING IN ONE-DIMENSIONAL ITERATIVE LOGIC NETWORKS, by W.L. Kilmer, March 1963. 26 pp. AFCRL-63-68</p> <p>The main purpose of this paper is to present some results on the problem of determining, for an arbitrary SITN cell definition, the various ways in which corresponding SITNs can switch from equilibrium to equilibrium, and equilibrium to cycle, following single x_i input value changes. These results are also applied to non-SITN models discussed earlier by the author in order to extend the present theory of switching dynamics for iterative systems.</p>	<p>UNCLASSIFIED</p> <ol style="list-style-type: none"> 1. Nonlinear Control Theory 2. Switching Theory 3. Mathematical Logic <p>I. Kilmer, W.L.</p>	<p>AF Cambridge Research Laboratories, Bedford, Mass. ON DYNAMIC SWITCHING IN ONE-DIMENSIONAL ITERATIVE LOGIC NETWORKS, by W.L. Kilmer, March 1963. 26 pp. AFCRL-63-68</p> <p>The main purpose of this paper is to present some results on the problem of determining, for an arbitrary SITN cell definition, the various ways in which corresponding SITNs can switch from equilibrium to equilibrium, and equilibrium to cycle, following single x_i input value changes. These results are also applied to non-SITN models discussed earlier by the author in order to extend the present theory of switching dynamics for iterative systems.</p>	<p>UNCLASSIFIED</p> <ol style="list-style-type: none"> 1. Nonlinear Control Theory 2. Switching Theory 3. Mathematical Logic <p>I. Kilmer, W.L.</p>	<p>UNCLASSIFIED</p>

AFCRL-63-68

On Dynamic Switching in One-Dimensional
Iterative Logic Networks

by
William L. Kilmer

Montana State College
Bozeman, Montana

Contract No. AF19(604)-6619

Project No. 5632

Task No. 563202

Scientific Report No. 4

March, 1963

Prepared
for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS

**On Dynamic Switching in One-Dimensional
Iterative Logic Networks***

William L. Kilmer

Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

ABSTRACT

A SITN is a cascade of identical finite automata such that the i^{th} automaton receives an x_i input from the outside world and a y_i input from its left neighbor, and produces a z_i output to the outside world and a y_{i+1} output to its right neighbor. We prove three main theorems: 1) For every integer k there is a cell definition such that a corresponding SITN either can or cannot switch from equilibrium to a cycling condition following a single x_i change according as $n \leq k$ or $n > k$, respectively; 2) there do not exist algorithms to tell whether or not a given cell definition admits of a SITN that can start from equilibrium and following a single x_i change either a) switch into a cycling condition, or b) put out a $z_i = 1$ during a switching transient; and 3) there do not exist algorithms to tell whether or not a given SITN cell definition must have every switching transient following a single x_i change from equilibrium either a) die out a bounded number of cells to the right of the change, or b) extend all the way to the SITN boundary. All theorems are proved constructively on finite-state diagrams, and 2) and 3) hinge on an embedding of Minsky's Post Tag system results into such diagrams. We conclude with several iterative network equivalence demonstrations.

*This research was supported by Air Force Cambridge Research Laboratories Contract No. AF19(604)-6619, under the auspices of The Montana State College Electronics Research Laboratory, Bozeman, Montana. The author is currently at the Research Laboratory of Electronics, M.I.T., on leave from Montana State College.

I. Introduction

We consider a concatenation of n identical logic cells as shown in Fig. 1. The i^{th} cell has associated with it the following: 1) a memory state variable s_i , with domain of values a_1, a_2, \dots, a_m ; 2) an external (i.e., outside world) input variable x_i , with domain b_1, b_2, \dots, b_p ; 3) lateral input and output variables y_i and y_{i+1} respectively, each with domain c_1, c_2, \dots, c_q ; and 4) an external output variable z_i , with domain d_1, d_2, \dots, d_r . We assume that the functions $z_i(x_i, y_i, s_i)$ and $y_{i+1}(x_i, y_i, s_i)$ are realized with zero time delay across the cells; and that the function $s_i(x_i, y_i, s_i)$ is realized with unit time delay within the cell. At time $t = 0$ the y_1, x_1, s_1 variables are all assigned arbitrary values from their respective domains, and for all $t > 0$ the values of y_1 and all the x_i remain fixed. We denote such cell systems SITNs (for Sequential IIterative Networks).

A SITN is said to be in equilibrium at $t > 0$ if and only if all of its y_i, s_i values remain fixed from t on. A SITN is said to be cycling at $t > 0$ if and only if its over-all configuration of y_i, s_i values at t first recurs at $t + T$, for $T > 1$. If a SITN is in equilibrium at $t = -1$, and at $t = 0$ some y_1 and/or x_1 values change, the corresponding sequence of y_i, s_i value changes is called a transient just in case the SITN reaches equilibrium at some $t > 0$. Otherwise the SITN enters a cycle.

The main purpose of this paper is to present some results on the problem of determining, for an arbitrary SITN cell definition, the various ways in which corresponding SITNs can switch from equilibrium to equilibrium, and equilibrium to cycle, following single x_i value changes. We also apply these results to non-SITN models discussed in Kilmer (1961), (1962A), (1962B) in order to extend the present theory of switching dynamics for iterative systems.

We claim that the results of this paper furnish new insights into the classical long-range order problems¹ of statistical mechanics, sociology, nonlinear control theory, neurophysiology,² and genetics.

1. Those problems that involve the derivation of long-range order patterns from short-range order relations.

2. In particular, the operational problem of the reticular formation in vertebrate nervous systems. This formation is always the central command and control center in such systems.

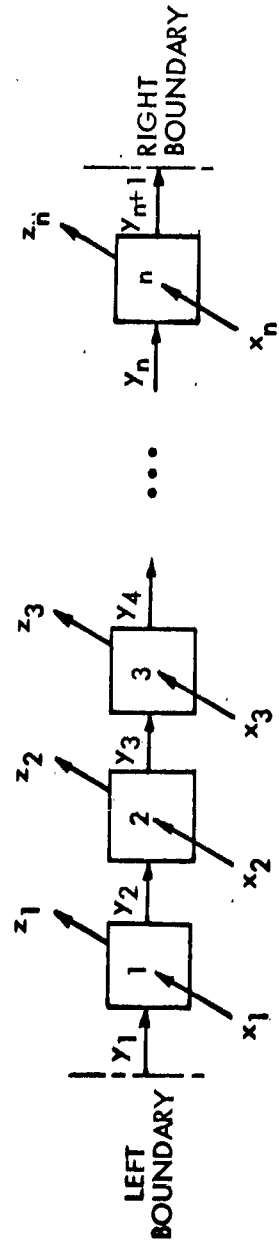


FIG. 1. SCHEMATIC DIAGRAM OF A SITN.

II. On the Relationship of SITN Size to Possible Cycle-Entry

Our concern in this Section is to give a constructive proof of Theorem 1.

Theorem 1: For every positive integer k there exists a SITN cell definition such that any corresponding n -celled SITN, which is in equilibrium at $t = -1$, and which has a single x_i value change at $t = 0$, either can or cannot possibly enter a cycle at some $t \geq 0$ according as $n > k$ or $n \leq k$ respectively.

Proof: Consider the partially complete³ SITN memory state diagram for $x_i = b_1$ shown in Fig. 2. The c_i/c_j label on each arrow there indicates that if a cell with x input equal to b_1 has the memory state value given at the tail of an arrow and if its y input value is c_i , its corresponding y output value is c_j and its next memory state value is the one given at the head of the arrow.

Now assume an n -celled SITN in equilibrium at $t = -1$ as follows:

all $x_i = b_1$; $s_i = a_1$ for i odd and $s_i = a_2$ for i even; and $y_1 = c_1$. This equilibrium is an obvious consequence of the self-returning arrows out of a_1 and a_2 . Then suppose that at $t = 0$, x_1 changes to b_2 . Figure 3 shows that this causes y_2 to change immediately from c_2 to c_1 . Figure 2 accordingly indicates the successive SITN variable value changes listed in Fig. 4. Each square's entry in Fig. 4 contains the column variable's value at the corresponding row time. Any column left blank above a certain row denotes that the column's topmost entry persists from that row time on. Note that the a_j values assumed by each s_i in Fig. 4 always have $(\text{maximum } j) = i$. Hence the $(k+1)^{\text{st}}$ cell is the leftmost one which can ever have its s_i variable take on the a_{k+1} value, and from that time on cycle around the a_{k+1}, a_{k+2} loop. Since this is the only cycle admitted by Fig. 2, the figure provides the essentials of a proof of Theorem 1.

We fill in the details of our proof in Fig. 5. Figures 5 and 3 clearly indicate that the only possible equilibrium s_i values are a_1 and a_2 . Thus it is easily seen that regardless of the sequence of x_i values along any corresponding SITN in equilibrium, if any single x_i value change is to cause the SITN to enter a cycle, it must do so essentially in accordance with Fig. 4. Hence at least

3. In the obvious sense that out of each s_i memory state value there should be an output arrow for each possible y_i value.

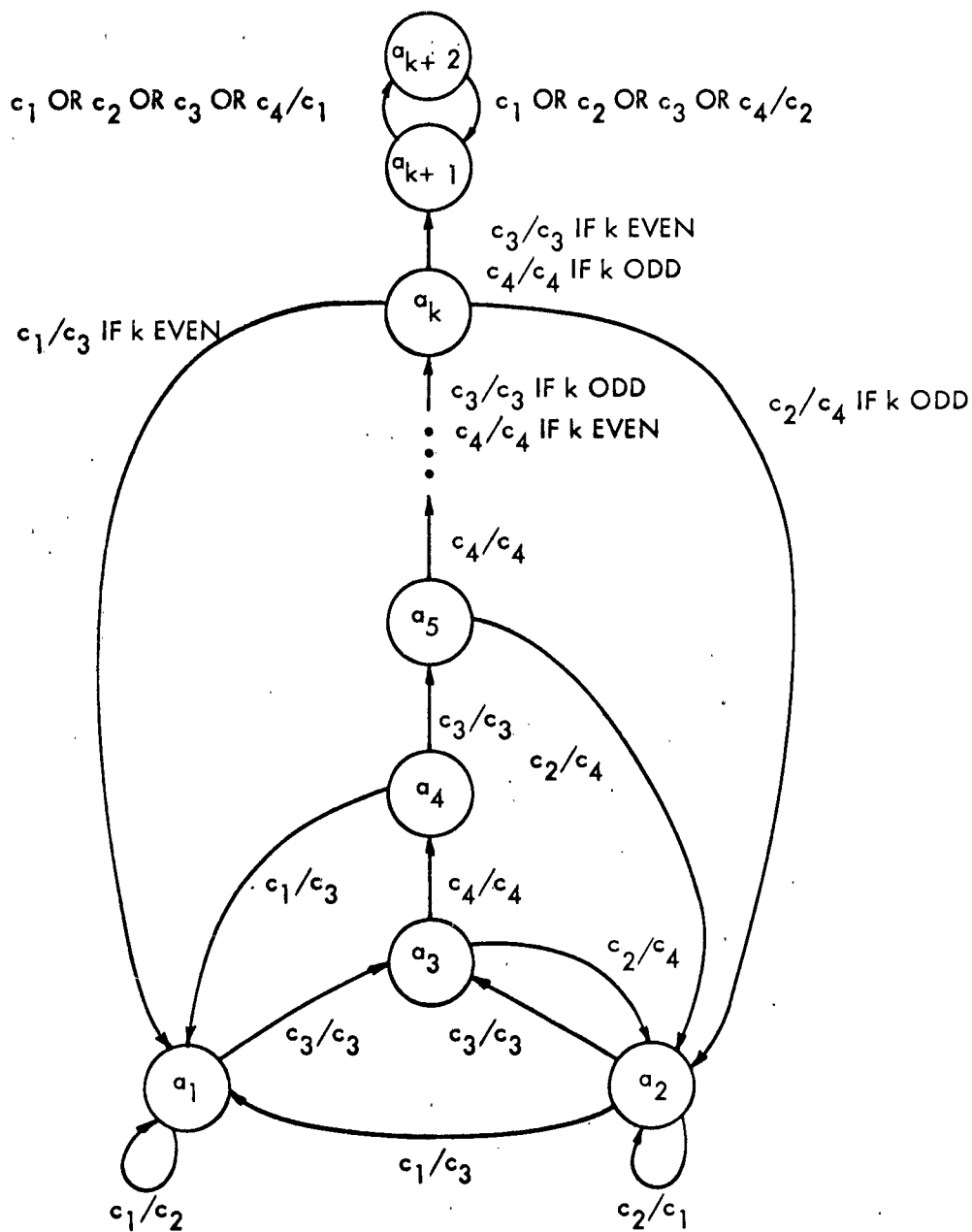


FIG. 2. PARTIALLY COMPLETE SITN MEMORY STATE DIAGRAM FOR $x_i = b_1$ USED IN THE PROOF OF THEOREM 1.

ALL THE REST OF THIS
MEMORY STATE DIAGRAM
(i.e., ALL BUT THE c_1 INPUT
ARROW OUT OF a_1 AND THE c_2
INPUT ARROW OUT OF a_2)
IS THE SAME AS THAT
FOR $x_1 = b_2$

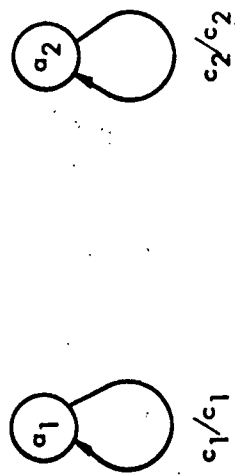
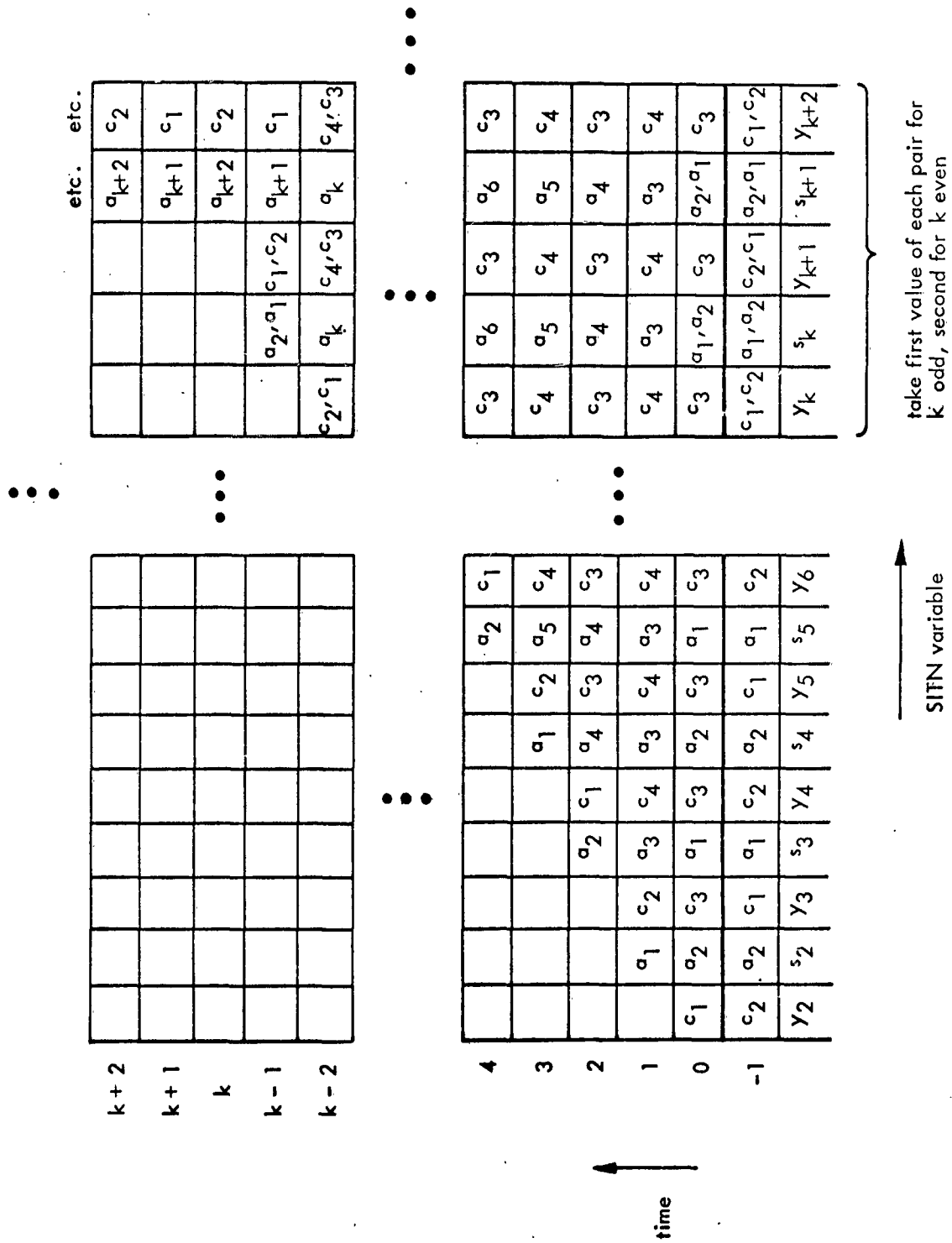


FIG. 3 SITN MEMORY STATE DIAGRAM FOR $x_1 = b_2$
USED IN THE PROOF OF THEOREM 1.



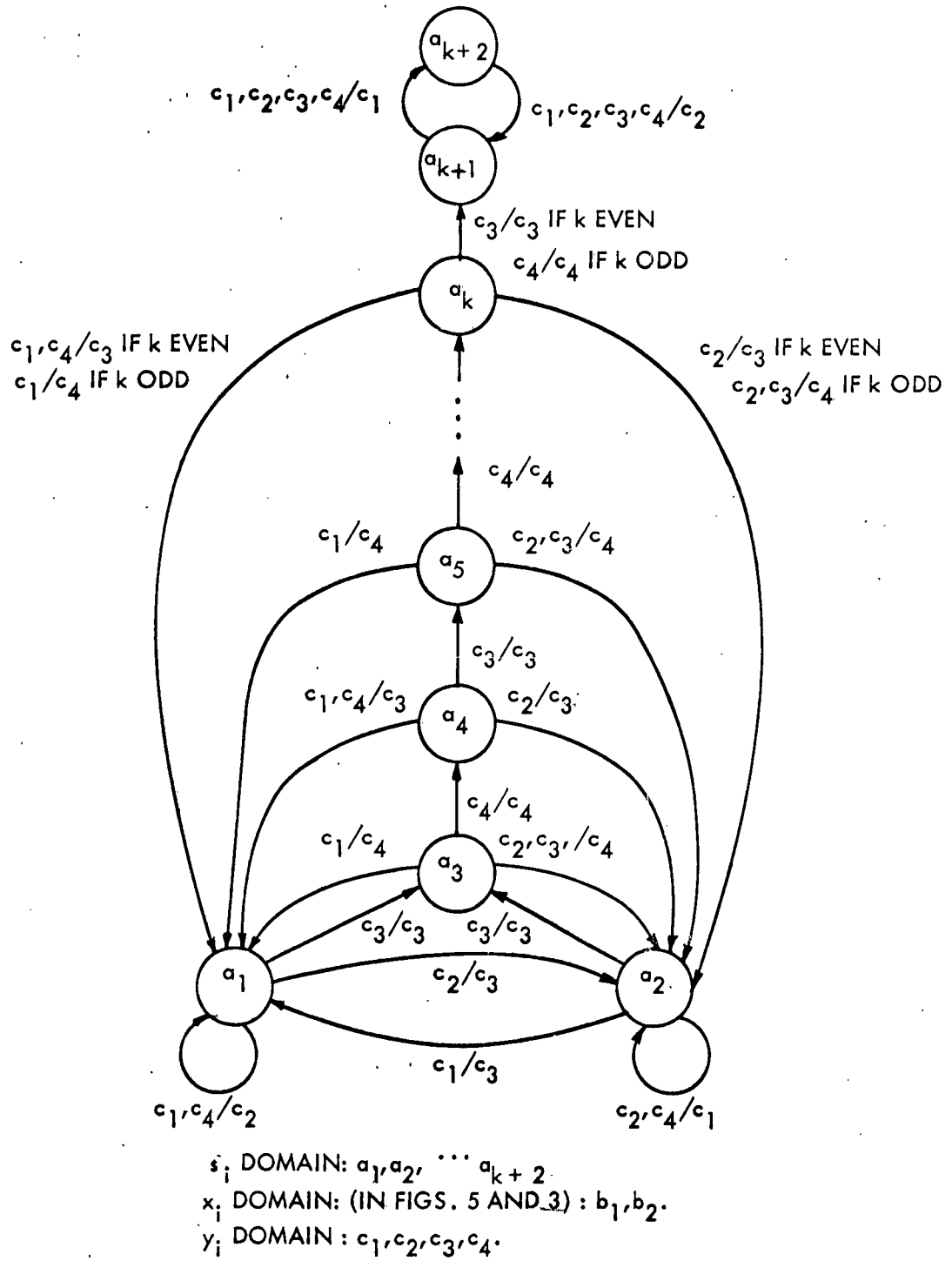


FIG. 5. COMPLETE SITN MEMORY STATE DIAGRAM TOGETHER WITH FIG. 3) USED IN THE PROOF OF THEOREM 1

$k + 1$ cells are always required to the right of any single x_i value change if a Fig. 5-type SITN is to enter a cycle under the conditions of Theorem 1.

Q. E. D.

III. Some Unsolvable Problems on Cycle Entry, Equivalence, and Transient Character

In this Section we prove two unsolvability theorems, using essentially one SITN memory state diagram and Minsky's (1961) result that universal Turing machines can be represented by Post tag systems. We state both theorems before proving either.

Theorem 2: There does not exist a recursive procedure to determine of an arbitrary SITN cell definition⁴ whether or not any corresponding SITN in any arbitrary equilibrium at $t = -1$ can have a single x_i value change at $t = 0$ cause it to:

- i) enter a cycle at some $t \geq 0$,
 - or ii) pass through a transient which causes a 1 output on some z_i at some $t \geq 0$.
- The ii) part of this Theorem pertains to the existence of certain SITN equivalence tests [cf. Hennie, (1961)].

We now consider SITNs which if in equilibrium at $t = -1$ and subjected to single x_i value changes at $t = 0$, admit only transient responses (i. e., no cycle entries at any $t \geq 0$). We call such SITNs transient SITNs. In case a transient-SITN cell definition is such as to insure that all single x_i changes from equilibrium cause transients involving y_i value changes all the way to the right boundary of every corresponding SITN, we call the cell definition boundary transient. And in case a transient-SITN cell definition is such as to insure that no single x_i change from equilibrium can cause transients involving y_i value changes more than a bounded (hence calculable) number of cells to the right of the x_i change in any corresponding SITN, we call the cell definition bounded transient.

Theorem 3: There does not exist a recursive procedure to determine of an arbitrary transient-SITN cell definition whether or not it is:

- i) boundary transient,
- or ii) bounded transient.

4. Such as given in Figs. 3 and 5 for example.

Our first step in proving Theorems 2 and 3 is to define a Post tag system. Let A be a finite set of letters a_1, a_2, \dots, a_m ; and let W be an associated set of words, such that for each i , W_i is a fixed string or word of letters of A . Let P be some integer, and define the following process applied to some initially given string S of letters of A : Examine the first letter of the string S . If it is a_i , remove the first P letters of S , and then adjoin the word W_i to the end of the remainder. Perform the same operations, defined a production, on the resulting string, and repeat so long as there are P or more letters left in each resulting string. If at some point there are fewer than P letters left in the resulting string, the process is said to terminate at that string. We call A, W, S, P , and the process just defined a Post tag system. Minsky, (1961), showed that the problem of determining for any given Post tag system whether or not the corresponding process ever terminates is recursively unsolvable. We will now embed this result into a SITN memory state diagram.

We replace the letters a_1, a_2, \dots, a_m (but not the symbols S, W_i) of a given Post tag system by the y_i values c_1, c_2, \dots, c_m respectively, and then complete the y_i domain by adding three special values, ϕ, ω_1 , and ω_2 . The latter two are interpreted as marker values, and ϕ is interpreted as the null value. Next we specify a_0, a_1, a_2, a_3 , and a_4 as the only possible equilibrium s_i values, and b_1, b_2 as the x_i domain. Figures 6 and 7 then give the essential outline of a SITN memory state diagram sufficient to prove Theorems 2 and 3.

Our notation in these Figures is as follows: C denotes any y_i value; $\sim\omega_1, \sim\omega_2$, and $\sim\phi$ denote any y_i values but ω_1, ω_2 , and ϕ respectively; y_i values raised to the j^{th} power denote j successive repetitions of those values; $\ell(T)$, for any string of letters T , denotes the number of letters in T ; and TU , T and U both strings, denotes the string consisting of the letters of T followed by the letters of U in order.

Let us now assume, in order to explain Figs. 6 and 7, that we have a corresponding SITN in equilibrium at $t = -1$ as follows: all x_i values are b_1 , s_1 is a_0 , s_2 is a_1 , and all other s_i are a_3 . Then suppose that at $t = 0$ there is a single x_i value change from b_1 to b_2 in the first cell. This causes y_2 to change from ϕ (i.e. null) to ω_1 . Subsequently s_2 passes from a_1 down through a_5 to a_2 , causing y_3 to put out the string $\omega_2 S$ before settling down at ϕ .

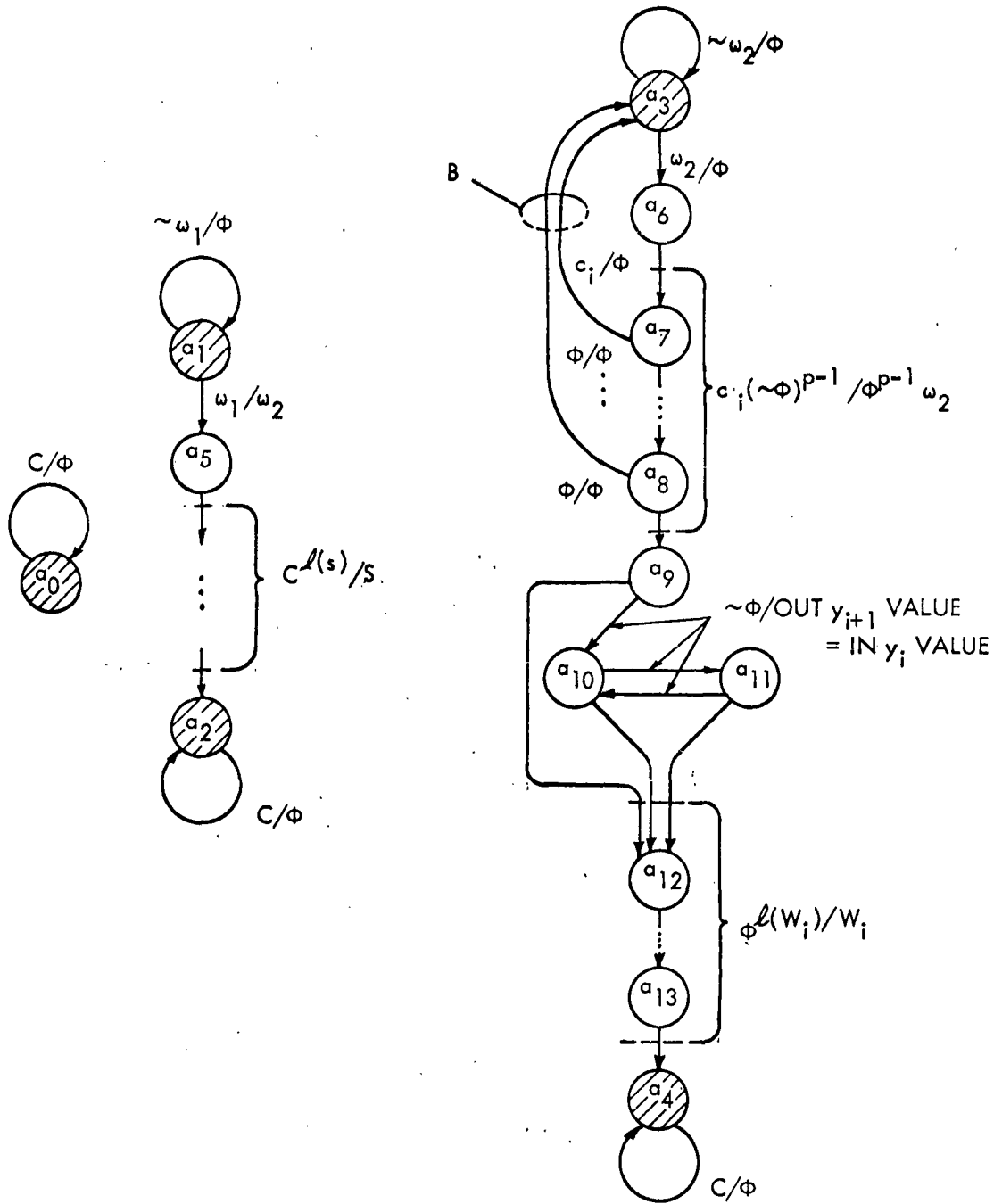


FIG. 6. THE PARTIALLY COMPLETE SITN MEMORY STATE DIAGRAM FOR $x_i = b_1$ USED IN THE PROOF OF THEOREMS 2 AND 3.



THE REST OF FIG. 7 IS THE SAME AS THE
CORRESPONDING PART OF FIG. 6, EXCEPT
 a_1 IS INTERCHANGED WITH a_2 , AND a_3
IS INTERCHANGED WITH a_4 .

FIG. 7. THE SITN MEMORY STATE DIAGRAM FOR $X_i = b_2$
USED IN THE PROOF OF THEOREM 2 AND 3

Next we show that y_4 accordingly puts out essentially the result of the first production in the corresponding Post tag system. To see this, we note that s_3 is taken from a_3 to a_6 by $y_3 = \omega_2$; from a_6 to a_7 by the next value of y_3 (i.e., the first letter of S , assumed c_1), and from a_7 to a_9 by the next $P-1$ values of y_3 (i.e., the next $P-1$ letters of S , whatever they might be). y_4 's value remains ϕ , or null, during all of these changes. Following them, however, the $(P+1)^{st}$ non- ϕ value of y_3 (i.e., the P^{th} letter of S) produces ω_2 out at y_4 . Then the sequence consisting of y_3 's $(P+2)^{nd}$ non- ϕ value to its last non- ϕ value (i.e., the $(P+1)^{st}$ to the last letter of S) produces itself out at y_4 . Finally, when y_3 settles down at ϕ , s_4 leaves a_{10} or a_{11} (whichever state it is in) and causes y_4 to put out the string W_i , corresponding to the first letter of S . After that y_4 also settles down at ϕ . Thus the first production in the Post tag system,

$$S = \underbrace{c_1 T_1}_{P \text{ letters}} \underbrace{T_2}_{\quad} \rightarrow T_2 W_i,$$

is represented by the $y_3 \rightarrow y_4$ transformation across the 3^{rd} SITN cell, $\omega_2 S \rightarrow \omega_2 T_2 W_i$ (preceding and succeeding ϕ values not shown).

More generally, the $y_i \rightarrow y_{i+1}$ transformation across the i^{th} SITN cell can be made to represent the $(i-2)^{nd}$ tag system production as follows: For each i in the tag system alphabet of letters, $\{c_i\}$, we add to Fig. 6 a portion of s_i state diagram which is exactly the c_i^{th} counterpart of the portion already there from a_7 to a_{13} . This is primarily to enable the first c_j value in each incoming y_i sequence to direct s_i to a portion of over-all state diagram that ends y_{i+1} 's non- ϕ sequence with the right W_j . The W_j are produced in one of the a_{12} - a_{13} -type portions of augmented s_i state diagram, and are SITN representations of completions of corresponding tag production steps. The purpose of adding a_7 - a_{11} as well as a_{12} - a_{13} -type portions of s_i state diagram is threefold: 1) the a_7 - a_8 portions enable the i^{th} cell to effectively remove (i.e., replace by ϕ) the 2^{nd} to P^{th} c_j values of each incoming y_i sequence. This begins the SITN representation of each corresponding tag production; 2) If there are ever fewer than P c_j values, the a_7 - a_8 portions allow y_{i+1} to remain fixed at ϕ . In each such case one Bundle arrow in Fig. 6 is traversed; and 3) The a_9 - a_{11} -type portions enable the i^{th} cell to simply pass the $(p+1)^{st}$

to last c_j values of each y_i sequence. Thus they represent intermediate tag production steps, preparatory to W_j adjoinments.

Hence if the corresponding tag system productions terminate at the i^{th} string, y_{i+2} is the leftmost SITN value that is left unchanged in the associated network transient.

We now finish our proof of Theorem 3 by filling in the missing details of Fig. 6 in Fig. 8. We leave it to the reader to check in Figs. 7 and 8 that only one type of transient can involve y_i value changes more than one cell to the right of a single x_i perturbation of a corresponding SITN at equilibrium. And that transient type is the one discussed above. Since the problem of determining whether or not the tag system productions corresponding to such a transient ever terminate is recursively unsolvable, so also is the problem of determining whether or not the transient itself is bounded or boundary. This completes our proof of Theorem 3.

The proof of Theorem 2 follows almost trivially. We prove part ii) by modifying the B bundle in Fig. 8 as follows: Instead of directing this bundle into a_3 , we direct it into a new state, a_{14} , as shown in Fig. 9. Then we specify that $z_i = 0$ for all s_i values except a_{14} , in which case $z_i = 1$. Theorem 2 ii) follows immediately by noting that it is yes if and only if the transient in the proof of Theorem 3 is bounded. But this question is recursively unsolvable.

We prove Theorem 2 i) by modifying Fig. 9 as shown in Fig. 10. Then states a_{14} and a_{15} comprise the only cycle that is accessible under the conditions of Theorem 2. Hence Theorem 2 i) is yes if and only if Theorem 2 ii) is yes, which is recursively unsolvable.

Q. E. D.

As a passing point, we note that by identifying the right and left boundary signals in the SITNs of Theorem 3, we can get a result much like Theorem 1, but with the inequalities reversed. Although this point has considerable interest, we will not develop it further here.

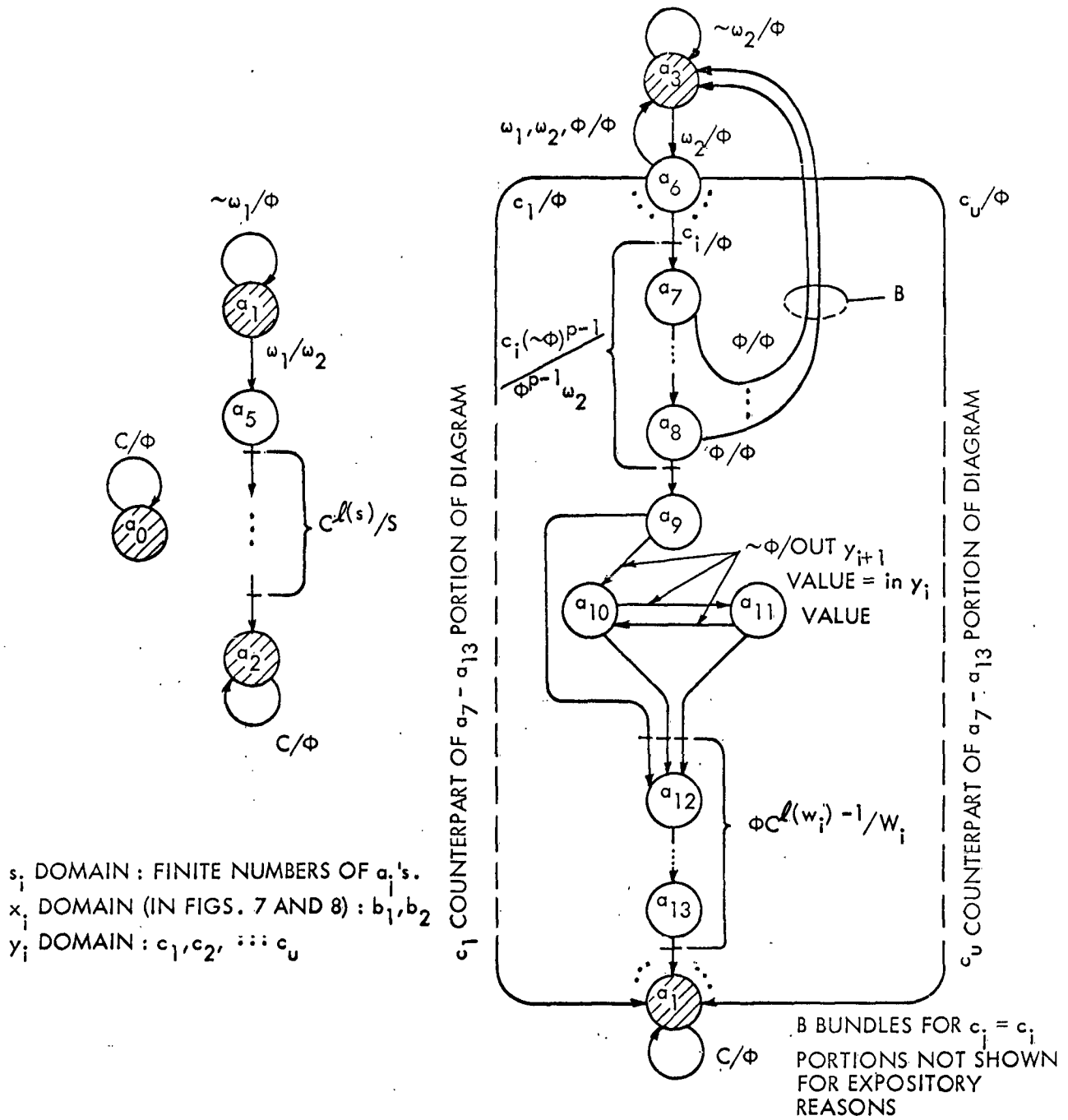


FIG. 8. COMPLETE SITN MEMORY STATE DIAGRAM TOGETHER WITH FIG. 3) USED IN THE PROOF OF THEOREM. 3

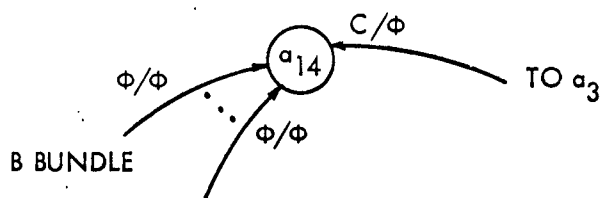


FIG. 9 . CHANGE IN FIG. 8 FOR
PROVING THEOREM 2ii)

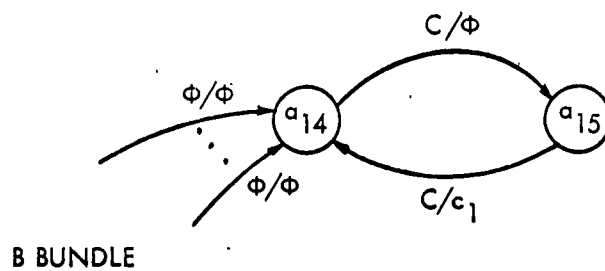


FIG. 10. MODIFICATION OF FIG 9 FOR
PROVING THEOREM 2i

IV. Applications to Non-SITN Models

In this Section we apply our results to some non-SITN models discussed in Kilmer (1961), (1962A), (1962B). Our method is to develop a chain of equivalences from one of those models to SITNs.

First, we define the network model shown in Fig. 11. The large square boxes there represent identical combinational logic cells, each having zero switching delay, and the small rectangles represent unit delay elements. Cellular α , β , and x inputs are constant during each unit time interval, so the network operates synchronously. Each cell's α_i and β_i lateral inputs and x_i external input take on values ranging over finite α_i , β_i , and x_i domains respectively. Correspondingly, each cell's α_i and β_i lateral outputs and z_i external output range over finite domains of values as determined by the truth table comprising the network's cell definition. We require only that the number of network cells be finite, and define such networks BITNs (for Bilateral ITERative Networks).

In Fig. 12 we show a reconception of Fig. 11. There the i^{th} cell maps $\alpha_i \rightarrow \beta_{i-1}$ under the influence of x_i and a left-coupling parameter, $C^{\ell i}$; and also maps $\beta_i \rightarrow \alpha_{i+1}$ under the influence of x_i and a right-coupling parameter, C^{ri} , all with zero delay. The coupling idea is to let $C^{\ell i}$ be a function of β_i such that all those β_i values which exert the same influence in every $\alpha_i \rightarrow \beta_{i-1}$ mapping cause the same $C^{\ell i}$ value. Similarly for C^{ri} and α_i values.

Fig. 11-to-Fig. 12-transformations are easily made 1-to-1. To illustrate, assume in Fig. 13 that the $\alpha_i \beta_i \alpha_{i+1}$ portion of the left-hand table is identical for all x_i values. Then β_i maps into α_{i+1} in the same way regardless of x_i 's value and whether α_i 's value is 0 or 2. Therefore let $C^{ri}(\alpha_i)$ be R_1 for $\alpha_i = 0$ or 2, and R_2 for $\alpha_i = 1$. Similarly for the right-hand table, assume that the $\beta_i \alpha_i \beta_{i-1}$ portion of the table is identical for all x_i values. Then let $C^{\ell i}(\beta_i)$ be L_1 for $\beta_i = 0$, and L_2 for $\beta_i = 1$ or 2. Our method should be clear by now, so we omit the remaining details.

Henceforth we denote Fig. 12 renditions of BITNs, BITN*s. And if the C^{ℓ} domain has only one element, we denote the corresponding networks R-BITN*s (for right-coupled BITN*s). Now consider the R-BITN* shown in Fig. 14. We note that each light dashed rectangle there encloses a structure

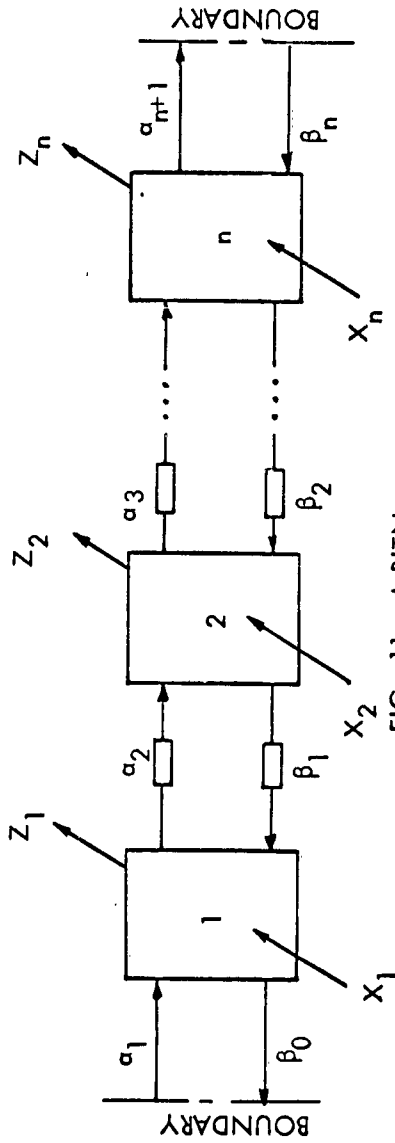


FIG. 11. A BITN.

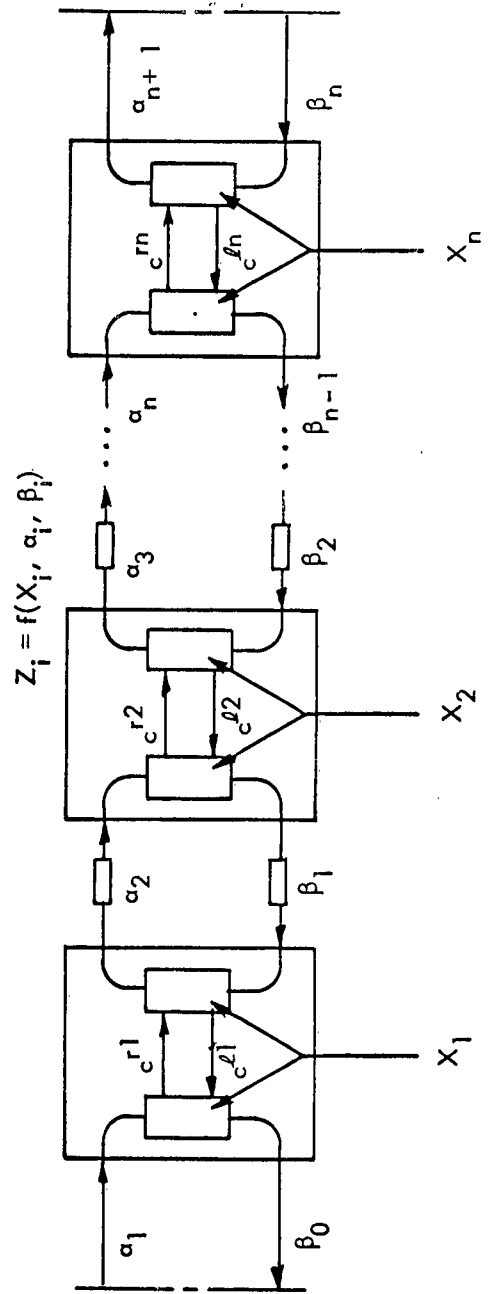
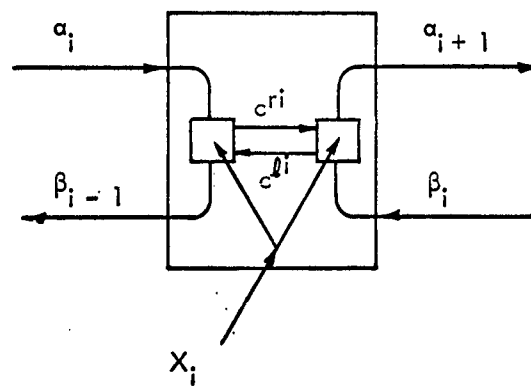


FIG. 12. A RECONCEPTION OF FIG. 11



X_i	a_i	β_i	a_{i+1}
0	0	0	1
0	0	1	2
0	0	2	0
0	1	0	1
0	1	1	2
0	1	2	1
0	2	0	1
0	2	1	2
0	2	2	0
1	0	0	1

\vdots

X_i	β_i	a_i	β_{i-1}
0	0	0	1
0	0	1	1
0	0	2	0
0	1	0	2
0	1	1	2
0	1	2	1
0	2	0	2
0	2	1	2
0	2	2	1
1	0	0	1

\vdots

FIG. 13. AN OUTLINE FOR A FIG. 11-TO-FIG. 12 TRANSFORMATION

which closely approximates a SITN cell. We will show that the network in Fig. 14 is, in fact, equivalent to a SITN.

Suppose in Fig. 14 that C^{r1} maps β_1 into α_2 at t . Then this α_2 produces C^{r2} , which maps β_2 into α_3 at $t+1$. This α_3 in turn produces C^{r3} , which maps β_3 into α_4 at $t+2$, and so forth. Thus if one knows C^{r1} at t , $t+2$, $t+4$, ..., and one knows β_1 into cell 1 at t , β_2 into cell 2 at $t+1$, ..., and β_n into cell n at $t+n-1$ for a R-BITN*, one has sufficient information to establish exactly half of its α and β values during each successive time interval. Hence the listed set of β 's and associated C^r 's is called the R-BITN*'s correspondence set at t . (We note that such a set is generally quite distinct from the analogous "initial condition set.")

Obviously any two independent R-BITN* correspondence sets, say at t and $t+2k+1$ for some integer k , respectively, are analyzed separately, yet in the exact same way, in order to determine their respective response. Each set is also analyzed independently of the unit time delay between cells. Hence the R-BITN* in Fig. 14 is equivalent to the SITN in Fig. 15 under the conditions that⁵:

1) the small rectangles beneath each cell in Fig. 15 represent unit delays; and

2) in Fig. 15 C^{ri} and β_i into cell i at time t map into $C^{r(i+1)}$ and β_i out of cell i at time t , just as in Fig. 14 C^{ri} and β_i of the correspondence set at t map into $C^{r(i+1)}$ and β_i of the correspondence sets at t and $t+2$, respectively.

From this discussion, we readily see Theorem 4.

Theorem 4: For each SITN result in Theorems 1, 2, and 3, there are exactly analogous results for R-BITN*s, BITN*s, and BITNs.

We remark that Theorems 3 and 4, with more or less immediate proof modifications, give strengthened versions⁶ of Hennie's (1961) Theorems 10, 10.1, 10.2, 11, 11.1, and 15. Also since the proof of Theorem 3 embodies a SITN representation of a universal Turing machine [cf. Minsky, (1961)], the result

5. Hennie, (1961), has developed a class of equivalence results that are related to, but essentially distinct from, those derived above.

6. Because we start from equilibrium instead of arbitrary initial conditions.

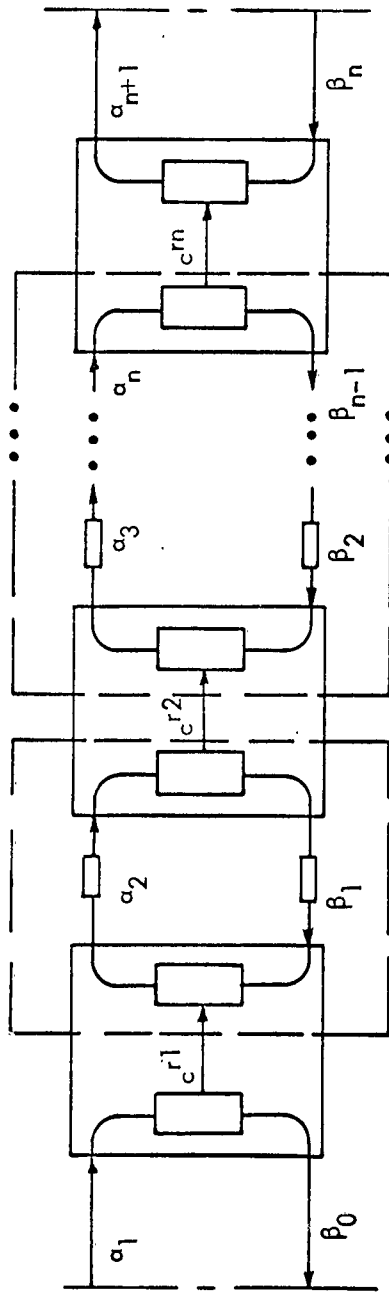


FIG. 14. AN R-BITN*

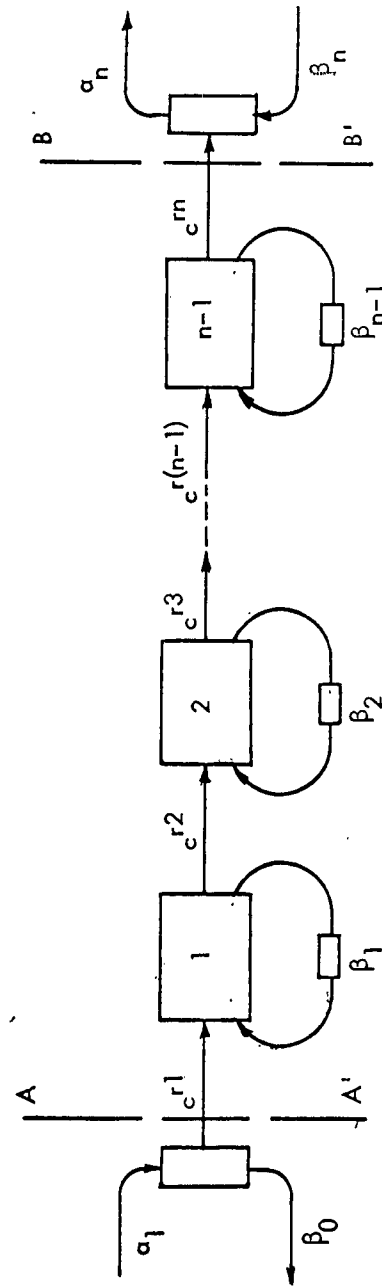


FIG. 15. A SITN EQUIVALENT OF AN R-BITN*

clarifies several computing capacity problems alluded to in Hennie, (1961). Finally, we claim that the present paper distributes the proof burden for Theorems 3 and 4 in such a manner as to substantially illuminate the basic nature of Hennie's previous work.

V. Conclusions

Kilmer, (1961), and Winograd, (1962), essentially closed out the main switching transients problems for BITNs which have either α_i or β_i lines missing. Hennie's previous work, (1961), extends these results to canonical decompositions of over-all memoryless BITNs (i.e., BITNs which have 1-to-1 over-all equilibrium $\{x_1, x_2, \dots, x_n\}$ input - $\{z_1, z_2, \dots, z_n\}$ output relations). Kilmer, (1962B), discusses the unsolvable nature of steady-state cycling problems in BITNs, BITN^{*}s, and R-BITN^{*}s. And this paper shows the essential unsolvability of the main transients problems in BITNs and SITNs. Thus future work must be directed at developing sufficiency conditions for desired transient behavior in such networks.⁷

In closing we note the curious duality between Theorem 1 of this paper and Theorem 2 of Kilmer, (1962A). The latter states: For every positive integer k , there exists a BITN cell definition such that every corresponding n -celled BITN is or is not over-all combinational according as n is or is not $\leq k$.

Now the curious thing about these theorems is that both of their (constructive) proofs seem to require cell complexities that are directly proportional in some sense to k . For Theorem 1 this proportionality is between k and the size of the s_j domain; and for Theorem 2 it is between k and the number of rows in the corresponding cellular truth table definition. The author is not sure what this really means in terms of recursive function theory, if indeed anything, but it certainly suggests a Cantor diagonalization approach.

7. (Note added in proof) At least one such set of conditions has already been derived, and it appears that others are forthcoming.

Acknowledgment

The author wishes to acknowledge Dr. Warren McCulloch, of the Research Laboratory of Electronics, M.I.T., as the one who demonstrated to him the importance of iterative network research to the neurophysiological study of reticular formations in vertebrate nervous systems. At present, the author's main motivations and guidelines are based on this awareness.

References

- Davis, M. "Computability and Unsolvability," McGraw-Hill, New York, 1958.
- Hennie, F. "Iterative Arrays of Logical Circuits," The M.I.T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, 1961.
- Kilmer, W. (1961). "Transient Behavior in Iterative Combinational Switching Networks," Proc. AIEE Symposium on Switching Theory and Logical Design, September, 114-128.
- Kilmer, W. (1962A). "Iterative Switching Networks Composed of Combinational Cells," IRE Trans. on Electronic Computers, EC-11, No. 2, April, 123-131.
- Kilmer, W. (1962B). "On Cycling Behavior in 1-Dimensional Bilateral Iterative Networks," Montana State College Electronics Research Laboratory Report, August.
- Minsky, M. (1961). "Recursive Unsolvability of Post's Problem of 'Tag' and Other Topics in the Theory of Turing Machines," Ann. Math., Vol. 74, No. 3, November, 437-455.
- Winograd, S. (1962). "Bounded-Transient Automata," Proc. AIEE Symposium on Switching Theory and Logical Design, September.